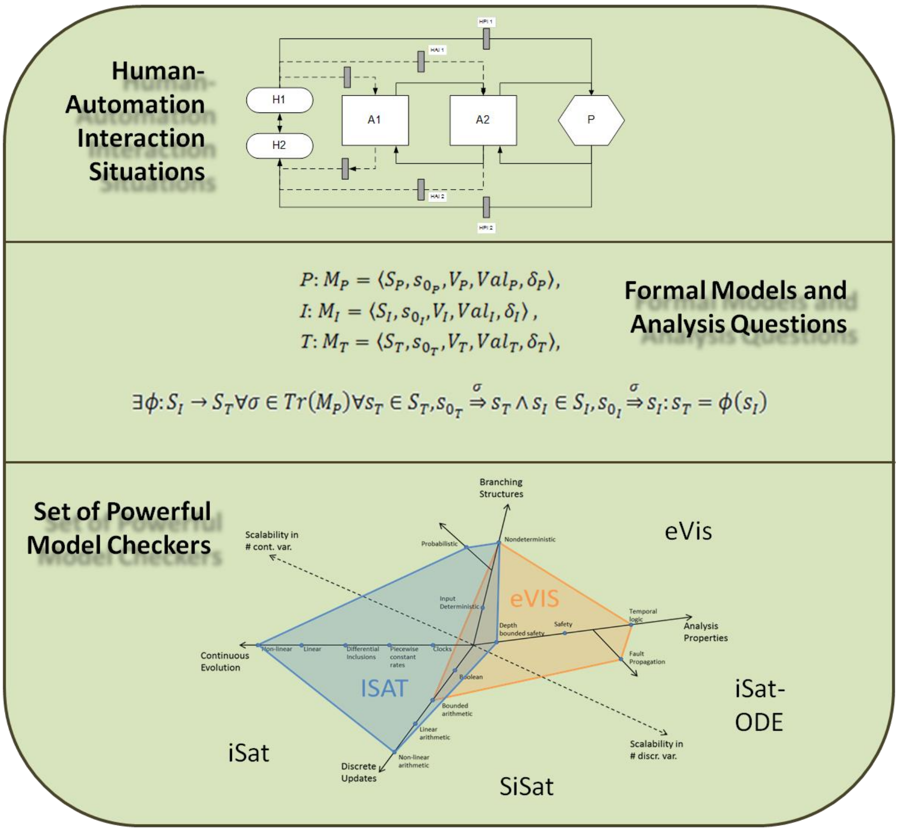
VASCO Verification Models for Advanced Human-Automation Interaction in Safety Critical Flight Operations

Executive Summary


Human-Automation Interaction Situations
HPI 1
HAI 1
H1
H2
A1
A2
P
HAI 2
HPI 2
Formal Models and Analysis Questions
$P: M_P = \langle S_P, s_{0_P}, V_P, Val_P, \delta_P \rangle,$
$I: M_I = \langle S_I, s_{0_I}, V_I, Val_I, \delta_I \rangle,$
$T: M_T = \langle S_T, s_{0_T}, V_T, Val_T, \delta_T \rangle,$
$\exists \phi: S_I \rightarrow S_T \forall \sigma \in Tr(M_P) \forall s_T \in S_T, s_{0_T} \overset{\sigma}{\Rightarrow} s_T \wedge s_I \in S_I, s_{0_I} \overset{\sigma}{\Rightarrow} s_I : s_T = \phi(s_I)$
Set of Powerful Model Checkers
Branching Structures
Scalability in # cont. var.
Probabilistic
Nondeterministic
eVis
Input Deterministic
eVIS
Temporal logic
Analysis Properties
Depth bounded safety
Safety
Non-linear
Linear
Differential Inclusions
Piecewise constant rates
Clocks
Continuous Evolution
Fault Propagation
Boolean
ISAT
iSat-ODE
Bounded arithmetic
Linear arithmetic
Scalability in # discr. var.
iSat
Non-linear arithmetic
Discrete Updates
SiSat

Authors

Author	Affiliation	Contact
Bertram Wortelen	OFFIS Institute for Information Technology	bertram.wortelen@offis.de
Sonja Sievi	Astrium Space Transportation GmbH	sonja.sievi@astrium.eads.net
Denis Javaux	Symbio	denis.javaux@symbio.pro

ESA Study manager: Maite Trujillo

This work was carried out under a program of and funded by the European Space Agency. The view expressed herein can in no way be taken to reflect the official opinion of the European Space Agency.

Objectives

Within VASCO the formal verification methodology (FVM) for advanced human-automation interaction in safety critical environments based on models of the overall human-automation interaction (HAI) system has been developed.

The main objectives for this work are summarized below:

- Approach human-automation interaction from a holistic cooperative system perspective where the object of design is a system of human agents (operators) and machine agents (automated systems) jointly performing a series of tasks (e.g., acquiring data, computing trajectories) aiming at satisfying higher level global task(s) (e.g., controlling a spacecraft).
- Seamlessly allow to consider human factor issues during system engineering in order to build systems that support strengths and compensate limitations of human behaviour characteristics.
- Address the knowledge gathered from other industries on the source and nature of human errors in advanced automated systems in a systematic way.
- Provide an extensive list of analysis questions for HAI situations.
- Consider human errors based on automatic human error injection into the nominal HAI system.
- Go beyond the analysis of prototypical HAI systems consisting of an operator, a user interface, an automation system and a controlled process to arbitrary complex systems with multi-level automation, multiple operators and interfaces.
- Evaluate the developed FVM on the basis of a case study.
- Propose recommendations for implementing the developed FVM into the design processes at ESA.

The project focusses on systems from the space domain. However, the FVM should be generic and should allow transferring knowledge and experiences from other domains into the space domain. Furthermore a special focus is put on human factor issues that are very specific to systems that include automated tasks.

Methodology

The study has been organized into the following work packages:

- Literature Review: Perform a literature review to derive an extensive database of analysis questions (AQDB) and a catalogue of approaches to address the questions;
- Verification Methodology Development: Develop the verification methodology which shall be able to answer all analysis questions of the AQDB.
- Case Study Development & Performance: Apply the FVM to a case study HAI system. Decompose, model and formalize the case study HAI system. Address analysis question with the formal model of the HAI system.
- Methodology Validation: Evaluate the practicability of the FVM based on the results and experiences of the case study performance.
- Conclusion and Recommendations: Reflect on the results of the FVM and the case study performance. Conclude on whether to include the FVM in design processes or not. Provide recommendations on how to implement the FVM in system design processes.

Results

Analysis Question Database

A database of 38 analysis questions has been collected based on a literature review. The questions are grouped in 7 categories. The questions focus on human factors aspects that are very important in the presence of automated systems. Each analysis question in the database is documented along a series of dimensions:

- the importance of the question (low, medium, high, very high);
- the cognitive stage(s) or step(s) the question is concerned with;
- the component(s) of the HAI situation the question applies to;
- existing (or non-existing) formal verification (FV) approaches or tools applicable or applied to address the question;
- alternative (non FV) approaches or methods than can be used to address the analysis questions, of particular interest when the FV approach is not or not easily applicable, e.g., expert reviews, questionnaires, HITL (Human In The Loop) simulation, virtual simulation;
- a comment that explains the question and its rationale.

All questions are listed below:

1) Information on Automation States and Behaviors

Information on automation states

C1.1: Is the operator informed on automation states?

C1.2: Is the information on automation state presented in such a way that it is (a) understandable and (b) unambiguous?

C1.3: Is the information on automation state sufficient to interact safely and efficiently with automation?

C1.4: Does a given action cause consistent effects?

Information on automation behaviours: Behaviour type 1 (state transitions)

C1.5: Is the operator informed when state transitions (e.g., mode transitions) occur?

C1.6: Is the information on automation behaviours presented in such a way that it is understandable and unambiguous?

C1.7: Is the operator informed on future state transitions (e.g., armed modes that will engage later)?

C1.8: Is the operator informed on the reason(s) for which state transitions occur (e.g., a mode reversion due to the loss of some system)?

Information on automation behaviours: Behaviour type 2 (effect of states on the process)

C1.9: Is the operator informed on the effects of automation on the process/plant/vehicle?

2) Issuing commands towards automation

C2.1: Is the operator informed by the user interface on the commands available to him/her (e.g., commands that cannot be used are dimmed, greyed,…)

C2.2: Is the translation of an operator goal into a series of commands on the UI a cognitively complex operation?

C2.3: Does the UI guide the operator in translating his/her goal into actions/commands on the UI?

C2.4: Has the operator the possibility to undo commands?

C2.5: Is the user able (or likely) to communicate goals to the system in an inappropriate order?

Feedback on operator actions

C2.6: Does a given action provide feedback?

C2.7: Is the feedback for an action delayed by the system?

3) Understanding automation: complexity issues

Planning actions on the UI for changing automation state (commanded changes)

C3.1: Is the translation of intentions into actions on the UI a simple operation?

Predicting automation behaviour (uncommanded changes)

C3.2: Is the structure of the state machine that underlies automation simple?

C3.3: Can the automation, as presented on the UI, be considered as a deterministic state machine for the operator

C3.4: Is the operator correctly informed on the internal states of automation and on the information it is processing (e.g., aircraft speed) to be able to predict if a state transition is going to occur and which one (e.g., uncommanded mode transitions due to sequencing, mode reversions due to parameters that near some threshold,...)

Building mental models of automation

C3.5: Can the state machine that underlies automation be abstracted into high-level rules that govern automation behaviour (e.g., all capture modes disengage when they have reached their target)?

C3.6: Are there few exceptions to these rules?

C3.7: Are state transitions that are exceptions to these rules announced in some way on the user interface (e.g., the "triple click" in the Airbus aircraft family).

Detecting abnormal situations

C3.9: Is the operator able to detect whether equipment or process is in abnormal mode; or whether there are latent abnormal conditions?

C3.10: Does the operator know when normal operating range is exceeded?

C3.11: Is the detection of abnormal situations a complex operation?

4) Situation Awareness and Out of the Loop problem

C4.1: Is all information needed to keep situation awareness to an appropriate level provided by the user interfaces when automation is in charge? Appropriate level meaning: sufficient to start performing the automated tasks rapidly if the automation stops doing them.

C4.2: Is support provided to the operator to rapidly build the appropriate level of situation awareness when the automation stops performing some tasks and the operator now has to take charge of them?

C4.3: Is transition from automation in charge to operator in charge "graceful", that is it is progressive, with some sub-tasks being progressively handled to the human operator?

5) Workload changes

C5.1: Are automated task re-distributions that increase human operator workload less likely during heavy operation phases (e.g., during the approach phase on a commercial airliner)?

C5.2: Is automation "aware" of the operator's workload and taking it into consideration to decide when to handle a task back to the operator?

6) Vigilance

C6.1: Are vigilance monitoring systems in place (e.g., based on gaze detection, iris size detection,...)?

C6.2: Are systems in place that allow keeping vigilance to a safe level (e.g., actions that operators have to perform for time to time, warnings that occur when vigilance gets below a given threshold,...)?

7) Skill acquisition / degradation

C7.1: Does the system provide margins for making errors, so the operator is able to experience and learn the system limits?

C7.3: Do operations (e.g., company policy, procedures,...) allow or force operators to regularly (e.g., once every 5 flights) manually perform the tasks that are automated?

8) Trust

C8.1: Is the behaviour of automation perceived by the operator as deterministic enough to for the automation to be trusted?

Formal Verification Methodology

The Formal Verification Methodology (FVM) developed in VASCO to address the analysis questions is a sequential stepwise approach. The steps of the FVM are shown in Figure 1.

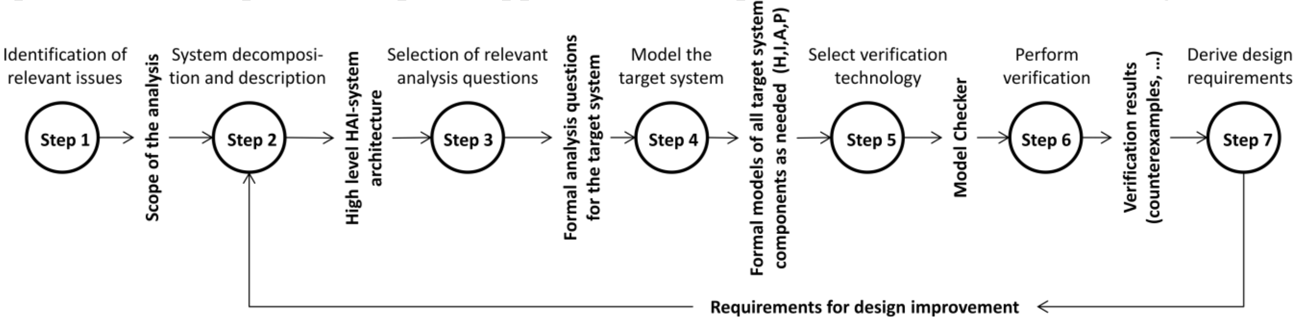

Identification of relevant issues
Step 1
Scope of the analysis
System decomposition and description
Step 2
High level HAI-system architecture
Selection of relevant analysis questions
Step 3
Formal analysis questions for the target system
Model the target system
Step 4
Formal models of all target system components as needed (H,I,A,P)
Select verification technology
Step 5
Model Checker
Perform verification
Step 6
Verification results (counterexamples, ...)
Derive design requirements
Step 7
Requirements for design improvement

Figure 1: Steps of the Formal Verification Methodology.

The objective of each step is as follows:

Step 1 Identify the relevant Human-Factors issues for the target Human-Automation Interaction system.

Step 2 Decompose the human-automation target system into the basic components (human agents, machine agents, tasks, processes, interfaces, environment).

Step 3 Select relevant analysis questions, amongst the 38 questions in the AQDB, so that they cover the Human-Factors issues identified in Step 1.

Step 4 Identify input models available in the design process for each component. If no input model is available for a component, then model it using adequate modelling techniques and associated editors.

Step 5 Select adequate formal verification techniques. The methodology provides guidance as to which verification techniques should be chosen dependent on the nature of the formal models (from Step 4).

Step 6 Perform verification of the analysis questions.

Step 7 Interpret the results of the verification and, if necessary, derive requirements for design improvements that counteract identified problems.

There are interdependencies between the steps, especially between steps 4, 5 and 6. For example, the kind of system model and the formalism used for the analysis questions restrict the choices for an adequate verification technique. But there are further constraints that restrict the choices like complexity issues or tool availability. This has to be considered when formalising the analysis question and modelling the system.

Improvements for the Case Study System

The FVM was evaluated on the basis of a case study system, that is the Environmental Control and Life Support (ECLS) system of the ISS module Columbus. The analysis was focussed on the air loop functionalities of the ECLS system. Each component of the system was formally modelled: The human operator, the user interfaces, the automation system and the controlled process (air flow).

With the FVM we were able to spot some weaknesses in the HAI system we were investigating. Following some findings are briefly described.

User Interface Design

With the technique of human error injection, we were able to automatically identify a weakness in the user interface design. Figure 3 shows an excerpt of Flight Procedure 2.102 which instructs the operator to execute a set of commands.

Based on the error model that we used, we showed that the likelihood of confusing the UI elements that are required to perform the procedure actions is (see Figure 2) is higher than normal. The procedure contains *verify* instruction in order to detect human errors. The analysis showed that the *verify* instructions are effected by the same root cause and thus will most likely not be able to detect this kind of human error. Several design improvement were proposed, like dividing the display shown in Figure 2 into two independent displays.

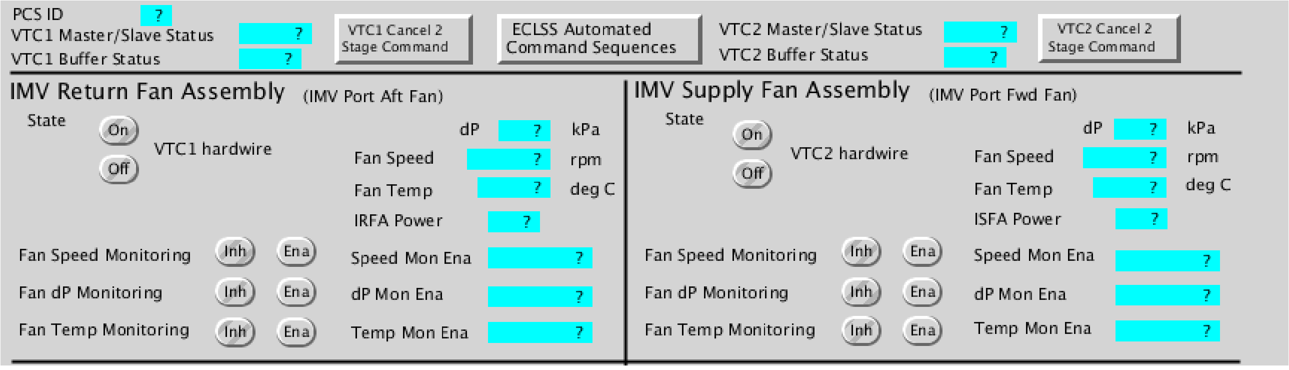
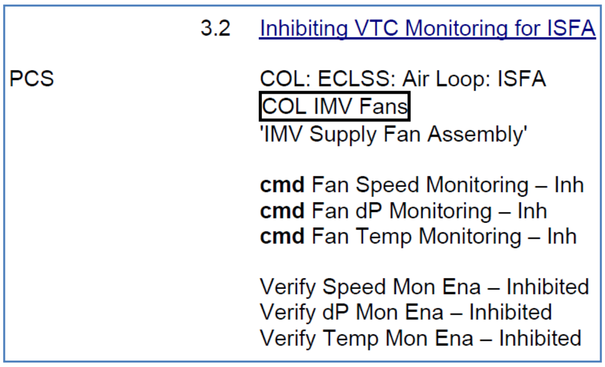

PCS ID ?
VTC1 Master/Slave Status ?
VTC1 Buffer Status ?
VTC1 Cancel 2 Stage Command
ECLSS Automated Command Sequences
VTC2 Master/Slave Status ?
VTC2 Buffer Status ?
VTC2 Cancel 2 Stage Command
IMV Return Fan Assembly (IMV Port Aft Fan)
State
On
Off
VTC1 hardwire
dP ? kPa
Fan Speed ? rpm
Fan Temp ? deg C
IRFA Power ?
Fan Speed Monitoring Inh Ena
Fan dP Monitoring Inh Ena
Fan Temp Monitoring Inh Ena
Speed Mon Ena ?
dP Mon Ena ?
Temp Mon Ena ?
IMV Supply Fan Assembly (IMV Port Fwd Fan)
State
On
Off
VTC2 hardwire
dP ? kPa
Fan Speed ? rpm
Fan Temp ? deg C
ISFA Power ?
Fan Speed Monitoring Inh Ena
Fan dP Monitoring Inh Ena
Fan Temp Monitoring Inh Ena
Speed Mon Ena ?
dP Mon Ena ?
Temp Mon Ena ?

Figure 2: Top part of PCS display 'COL IMV Fans'



3.2 Inhibiting VTC Monitoring for ISFA

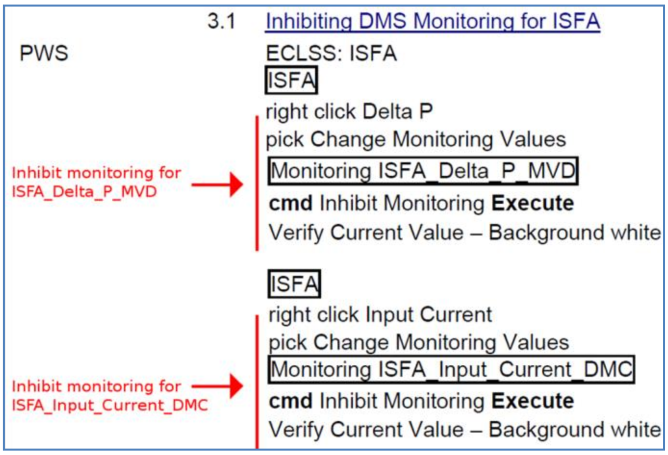
PCS

COL: ECLSS: Air Loop: ISFA
COL IMV Fans
'IMV Supply Fan Assembly'

cmd Fan Speed Monitoring – Inh
cmd Fan dP Monitoring – Inh
cmd Fan Temp Monitoring – Inh

Verify Speed Mon Ena – Inhibited
Verify dP Mon Ena – Inhibited
Verify Temp Mon Ena – Inhibited

Figure 3: Cmd Callouts to inhibit fan speed monitoring for ISFA in procedure 2.102.



3.1 Inhibiting DMS Monitoring for ISFA

PWS

ECLSS: ISFA
ISFA
right click Delta P
pick Change Monitoring Values

Inhibit monitoring for ISFA_Delta_P_MVD

Monitoring ISFA_Delta_P_MVD
cmd Inhibit Monitoring **Execute**
Verify Current Value – Background white

ISFA
right click Input Current
pick Change Monitoring Values

Inhibit monitoring for ISFA_Input_Current_DMC

Monitoring ISFA_Input_Current_DMC
cmd Inhibit Monitoring **Execute**
Verify Current Value – Background white

Figure 4: Step 3.1 of proc. 2.102 with two inhibit monitoring instructions

Flight Procedure Design

Error injection revealed a further weakness. This one regards the procedure design. The procedure excerpt in Figure 3 shows the task of inhibiting monitoring for a set of sensor values. The successful execution of all commands is checked at the end by a set of verify instructions. Some procedures have a different design for the same kind of task. Figure 4 shows a procedure, in which the *verify* instructions always follow directly after the *inhibit monitor* command. Both designs are in principle prone to errors of omission. However, the verification revealed that the likelihood of not detecting such an error is higher for the design in Figure 4.

The analysis of questions C1.4 and C3.3 revealed that executing Flight Procedure 2.102 leads to inconsistent results, depending on the initial system state in which the procedure is executed. No description of valid initial states is given at the beginning of the procedure. As result of the analysis it was proposed to provide such a description for the procedure, because it would help the operator to anticipate unexpected system behaviours for system states which are not specified as valid initial states.

Design of the socio-technical system that develops the HAI system

The unexpected inconsistent behaviour of procedure 2.102 can be solved by verifying a valid initial state at the beginning of the procedure as described above. However, design improvements can be made on an even higher level. In order to avoid introducing the same problem into future procedures we proposed to improve the socio-technical system (including the development team, managers, etc.) that develops HAI systems. This can for example be done by improving artefacts like checklists that check, whether the valid initial states of a procedure has been considered during procedure design.

Conclusion

From the results and evaluation of the study we draw the following conclusions:

- The FVM method is able to support the development of a well-designed HAI system.
- Formal verification is applicable to the analysis questions in the AQDB but some topics, to be investigated deeply, would probably require to rely on complementary methods (virtual simulations, HITL simulations, expert reviews, ...). See TN4, Step 1.
- The ability to proceduralize and/or automate the steps of the FVM is variable. It should be possible top proceduralize or even partly automate Steps 2, 4, and 5. Especially Step 6 is amenable for high automization. Though Steps 1, 3 and 7 can only more or less be performed "manually" in most cases, with limited proceduralization. These steps still require high level of human expertise (SE, OE, HFE) than today cannot be automated (though guidelines can be provided).
- The ability to proceduralize and/or automate the step is also dependent on the target human-automation interaction (HAI) system/object and human-automation interaction system description (Step 2). For example if the focus is on procedures (with at least one system and one human agent), it should be possible to automate a lot.
- Exploiting the result of the formal validation (Step 6) in Step 7 to derive design improvements is not easy and still requires combined expertise of an operations expert (OE) human factors experts (HFE) and system experts (SE), in a single individual or in a team. We recommend to develop tools that support this step by helping to analyse the traces produced as counterexamples by the model checker.
- To efficiently integrate the FVM in system design processes we highly recommend resorting to a model-based design approach. This will help (but not completely) addressing the difficulties at Steps 4 and 6. The models used for designing and specifying the HAI system should include all the information needed for their later verification. For example a model of some system modes explicitly stated in the design model. Or specific statements (contract-based approach) about the excepted output/behaviours of that system. The added benefit is also that these topics are brought in very early and in a very explicit way, so that they can be discussed and decided upon.

- The FVM can be introduced at different stages in the project life cycle:
    - The highest benefits can be gained in projects that make high usage of models, or follow the MBSE approach. In general we think that transition from phase B (definition) to phase C (design) is an early phase for introducing the FVM into a space project. That way human interaction, human automation interaction models and task models can be introduced and engineered together with system models. Formal verification can also be used for AIT. In this scope it could support the design and verification of integration and test procedures, answer if the whole system model is covered by integration and tests and additionally the consistency of the tests could be checked automatically.
    - A second entry point for introducing the FVM into ECSS phases could be in the transition of phase C (design) to phase D & E (production, utilization) when the operational concepts will be implemented. In this phase task and error models could directly be derived from potential use case models (e.g. from SysML, UML). Additionally task and error models could support the design of synoptic displays and procedures. As for AIT, the same applies for synoptic displays and operations procedures. Formal verification could also support the design, verification and consistency checks of the procedures and displays.
    - The third useful entry point is the phase E (utilization), FVM is then applied on a system in operations. Here in principle the same applies than when introducing FVM during implementation of the operational concepts. Since this is entry point is rather late, a trade-off has to performed between the availability of models and the ability of integrating synoptic displays and procedures into the models versus the time period of operations ahead.
- Corresponding organizational changes should be installed, so that dedicated teams integrating engineers (SE), Human Factors specialists (HFE) and operational specialists (OE) work together and design these human-automation interaction (HAI) systems. The socio-technical nature of these joint human-automation systems must be acknowledged and correctly dealt with by the institution that designs and verifies them.

Acronyms

Acronyms	Description
AIT	Assembly Integration and Test
AQDB	Analysis Question Database
ECSS	European Cooperation for Space Standardization
FV	Formal Verification
FVM	Formal Verification Methodology
HAI	Human-Automation Interaction
HFE	Human Factors Expert
HITL	Human In The Loop
MBSE	Model-Based System Engineering
OE	Operations Expert
SE	System Expert
UI	User Interface
VASCO	Verification Models for Advanced Human-Automation Interaction in Safety Critical Flight Operations